

# **EXHIBIT A**

## CISCO'S PROPOSED CONSTRUCTIONS

### I. The '526 Patent

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>"management programs"</p> <p><u>Proposed construction:</u></p> <p>"separate tools or external agents having their own respective command formats that provide management functions"</p>	<p>The specification of the '526 patent, including:</p> <p>"The parser, upon identifying a best match among the elements, issues a prescribed command for a selected one of the management programs according to the corresponding command format based on the selected command action value. Hence, a user may control multiple management programs having respective command formats, by using a set of generic commands that are independent from the command formats, eliminating the necessity that the user needs to learn the detailed command formats and syntax." (Abstract)</p> <p>"The parser, upon identifying a best match among the elements, issues a prescribed command for a selected one of the management programs according to the corresponding command format based on the selected command action value. Hence, a user may control multiple management programs having respective command formats, by using a set of generic commands that are independent from the command formats, eliminating the necessity that the user needs to learn the detailed command formats and syntax." (1:54-63)</p> <p>"One aspect of the present invention provides a method in a processor-based system configured for executing a plurality of management programs according to respective command formats." (1:64-68)</p> <p>"The method also includes issuing a prescribed command of a selected one of the management programs according to the corresponding command format, based on the identified one element." (2:8-12)</p> <p>"Another aspect of the present invention provides a system configured for executing a plurality of management programs according to respective command formats." (2:13-15)</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p data-bbox="659 321 1239 548">“The system also includes a plurality of translators configured for issuing commands for the management programs according to respective command formats, the parser outputting a prescribed command to a selected one of the translators based on the identified one element.” (2:24-29)</p> <p data-bbox="659 562 1239 1230">“As shown in FIG. 1, the management programs 18, implemented for example by different OAM tools such as RTM programs, may be executed within the processor based system or externally as external agents accessible using a prescribed application programming interface (API). The management programs 18 may provide different administration and maintenance functions, for example initiating various real-time screens used to monitor the internal state of executable processes within the software based system 10; alternately, different tools 18 may allow the user to control the various states within the various component of the software based system 10 via external programs (e.g., programs 18 c or 18 d), or may be used to issue external alarms (e.g., SNMP manager scripts) for external routines such as message waiting indicator routines.”</p> <p data-bbox="659 1245 768 1276">(3:1-15)</p> <p data-bbox="659 1291 821 1323">Figures 1, 3.</p> <p data-bbox="659 1337 1211 1436">The prosecution history of the ‘526 patent, including prior art references cited therein, including:</p> <p data-bbox="659 1451 1239 1707">“Pratt teaches a command parser in a computer system that recognizes commands and performs specific actions based on the parsed commands. However, Pratt teaches the device more as a natural language parser, and not a command system for management programs.” (File History, Rejection, Jan. 15, 2003)</p> <p data-bbox="659 1722 1195 1940">“Although Pratt discloses a method and apparatus for parsing commands using a parse tree, the Official Action fails to identify the presence of any <u>management programs</u> having respective command formats, where the generic command is mapped to a prescribed command for a</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p><u>selected</u> management program having a <u>corresponding command format</u>. In particular, the Official Action merely states in paragraph 2, page 2, that Pratt teaches ‘issuing a prescribed command based on the matched element (Column 1, lines 26-31).’” (File History, Applicant Response, Apr 16, 2003)</p> <p>“Scheiber et al. does not disclose ‘receiving a generic command from a user’, let alone ‘issuing a prescribed command of a <u>selected</u> one of the management programs according to the <u>corresponding command format</u>, based on the identified one element,’ as specified in claims 1 and 14. Rather Scheber [sic] et al. disclose a method allowing a user to control a <u>single computer application</u> at a time using spoken commands.” (File History, Applicant Response, Oct 7, 2003)</p> <p>“Finally, Shieber neither discloses <u>selection</u> of a management program, as claimed. As described above, the generic command provides an abstraction of command formats on a management program basis; hence, the command is for the <u>selected management program</u> according to the <u>corresponding selected command format</u>, based on the identified element. The Official Action also fails to identify the presence of any <u>management programs</u> having respective command formats, where the generic command is mapped to a <u>prescribed</u> command for a <u>selected</u> management program having a <u>corresponding command format</u>.” (File History, Applicant Response Oct 7, 2003)</p>	
<p>“generic command”</p> <p><u>Proposed construction:</u> “command that provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue the command based on the relative functions, as</p>	<p>The specification of the ‘526 patent, including:</p> <p>“Generic command interface for multiple executable routines” (Title)</p> <p>“A processor based system having a parser is configured for validating a generic command received from a user relative to a command parse tree. The command parse tree includes multiple elements, each</p>	<p>Generic: 1 a: relating or applied to or descriptive of all members of a genus, species, class, or group: common to or characteristic of a whole group or class: typifying or</p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
opposed to the specific syntax for a corresponding tool"	<p>specifying at least one corresponding generic command component and a corresponding at least one command action value. The parser, upon identifying a best match among the elements, issues a prescribed command for a selected one of the management programs according to the corresponding command format based on the selected command action value. Hence, a user may control multiple management programs having respective command formats, by using a set of generic commands that are independent from the command formats, eliminating the necessity that the user needs to learn the detailed command formats and syntax." (Abstract)</p> <p>"Hence, a user may control multiple management programs having respective command formats, by using a set of generic commands that are independent from the command formats, eliminating the necessity that the user needs to learn the detailed command formats and syntax." (1:58-63)</p> <p>"In particular, the parser <b>14</b> and the translators <b>16</b> provide a generic command syntax that integrates the functionality of the different tools <b>18</b> and that automatically selects the appropriate command for the best tool for executing a given generic command. As illustrated in Part A of the attached appendix, the new syntax provides a generic instruction set that provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue command based on the relative functions, as opposed to the specific syntax for a corresponding tool <b>18</b>." (3:27-35)</p> <p>"The parser <b>14</b> is configured for validating a received generic command by comparing each input command word to the command parse tree <b>22</b> to determine for the received generic command a tree element <b>24</b> identified as a best match." (3:47-51)</p> <p>"In particular, the parser <b>14</b> recursively traverses the command parse tree <b>22</b> for each command word to identify the best</p>	<p>subsuming: not specific or individual: GENERAL</p> <p>Webster's Third New International Dictionary, 2002</p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>match for the generic command. If only a portion of the generic command is identified as valid (e.g., only the first three command words are valid), the parser<sup>14</sup> selects the command key <b>32</b> for the matching token <b>28</b> from the last valid tree element <b>24</b>.” (3:54-61)</p> <p>Figures 1, 2, 3.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein, including:</p> <p>“The Official Action asserts that the term ‘generic’ can be reasonably interpreted to represent ‘any kind of command that <u>belongs to a general group or class.</u>’ This assertion is both without foundation, and logically inconsistent with well-known definitions. The proposed interpretation suggests that <u>any command that is a member</u> of a general group or class can be termed ‘generic.’ However, this interpretation does not describe a ‘generic’ command, but in fact describes a <u>specific</u> command belonging to a general group. In fact, Webster’s Dictionary, third College Ed., at page 562 (attached as Exhibit A) defines generic as ‘<u>of, applied to, or referring to a whole</u> kind, class, or group; inclusive or general....’ Hence the one skilled in the art would recognize that the term ‘generic’ relates to not a <u>member</u> of a class, but the <u>common attribute</u> of that class. Hence, the assertion that ‘the pre-interpreted command represents a generic command in that it <u>belongs to</u> [sic] an overall set of commands used to provide functionality to some program’ is nonsensical, inconsistent with the specification, and inconsistent with the use of the term ‘generic’ by one skilled in the art. There is <u>no</u> indication in Shieber et al. suggesting the command can be considered generic to any common attribute of a class. Finally, the Examiner’s interpretation of ‘generic’ is inconsistent with the <u>explicit claim language</u> that specifies: the generic command is validated based on command parse tree, and one of the elements identified as a best match relative to the generic</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>command is used to issue a <u>prescribed command of a selected management program.</u>" (File History, Response after Final Rejection, Feb. 18, 2004)</p> <p>"Shieber does not teach that the command is a generic command. Belknap, however, does teach inputting a generic command into a command parser, which then applies a specific command to a specific media device (Column 1, lines 45-57). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of receiving a command from the user, validating the command based on a command parse tree, and issuing a prescribed command based on the matched element, as taught by Shieber, where the command is a generic command, as taught by Belknap, since this allows a level of abstraction for specifying commands for a plurality of programs with different command formats." (File History, Rejection, March 18, 2004)</p> <p>"Hence, the specification describes that the parser 14 and the translators 16 provide a generic command syntax (e.g., a generic instruction set illustrated in Part A of the appendix) that <u>provides an abstraction of the tool-specific command formats and syntax.</u> Hence, to respond to the Examiner's query, a command with a 'prescribed command format' is not inherently or even necessarily a specific command, but rather can still be an abstraction of the tool-specific command format and syntax." (File History, Applicant Response, June 23, 2004) (emphasis in original)</p> <p>"Part A of the Appendix specifies numerous examples where a prescribed functional item ("Functional Item") having a tool-specific command format and syntax ("Old Command Line/Syntax") is replaced with a "New Syntax" that is an abstraction of the tool-specific command format and syntax" (File History, Applicant Response, June 23, 2004)</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>“Hence, multiple functions can be mapped to new syntax commands. Also note that the fact that execution of a generic command ‘watch acb globals’ may initiate execution of multiple functional items may be deemed by one skilled in the art an acceptable by-product of using the generic commands.” (File History, Applicant Response, June 23, 2004)</p> <p>“Part B of the attached appendix illustrates examples of generic commands, where each generic command has a corresponding identified usage (i.e. ,syntax) with prescribed acceptable parameters: Watch &lt;Object&gt; [Screen] // Get &lt;Variable&gt; // Set &lt;Variable&gt;&lt;Value&gt; // Start &lt;Agent&gt; (etc.)” (File History, Applicant Response, June 23, 2004)</p>	
<p>“command parse tree”</p> <p><u>Proposed construction:</u> “a hierarchical data representation having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value”</p>	<p>The specification of the ‘526 patent, including:</p> <p>“A processor based system having a parser is configured for validating a generic command received from a user relative to a command parse tree. The command parse tree includes multiple elements, each specifying at least one corresponding generic command component and a corresponding at least one command action value.” (Abstract)</p> <p>“These and other needs are attained by the present invention, where a processor based system having a parser is configured for validating a generic command received from a user relative to a command parse tree. The command parse tree includes multiple elements, each specifying at least one corresponding generic command component and a corresponding at least one command action value.” (1:48-54)</p> <p>“The method includes receiving a generic command from the user, and validating the generic command based on a command parse tree that specifies valid generic commands relative to a prescribed generic command format, the command parse tree</p>	<p>A parser reads the user’s source code programs and determines the <i>syntactic category</i> (part of speech) of every source symbol and combination of symbols. Its output is the list of the symbols defined in the program and a <i>parse tree</i>, which specifies the role that each source symbol is serving, much like a sentence diagram of an English sentence.</p> <p>Alice E. Fischer et al., THE ANATOMY OF PROGRAMMING LANGUAGES 74-75 (1993)</p> <p>A parser is a group</p>



Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating step including identifying one of the elements as a best match relative to the generic command.” (1:67-2:8)</p> <p>“The system includes a parser having a command parse tree configured for validating a generic command received from a user, the command parse tree configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the parser identifying one of the elements as a best match relative to the generic command.” (2:15-24)</p> <p>“FIG. 2 is a diagram illustrating in detail the parser <b>14</b> of FIG. 1 according to an embodiment of the present invention. The parser <b>14</b> includes a command word translation table <b>20</b> and a command parse tree <b>22</b>.” (3:36-39)</p> <p>“The parser <b>14</b> is configured for validating a received generic command by comparing each input command word to the command parse tree <b>22</b> to determine for the received generic command a tree element <b>24</b> identified as a best match. Each tree element <b>24</b> includes at least one token-command key pair <b>30</b> that specifies a token (T) <b>28</b> and a corresponding command key (CK) <b>32</b>, enabling the parser <b>14</b> to identify the appropriate prescribed command based on the command key specified for the matching token. In particular, the parser <b>14</b> recursively traverses the command parse tree <b>22</b> for each command word to identify the best match for the generic command.” (3:47-57)</p> <p>“The parser <b>14</b> then traverses the command parse tree <b>22</b> in step <b>42</b> to search for the matching token <b>28</b>. As illustrated in FIG. 2, the parser <b>14</b> locates the matching token in</p>	<p>of subroutines that converts a token stream into a parse tree, and a parse tree is a structural representation of the sentence being parsed.</p> <p>Allen I. Holub, COMPILER DESIGN IN C (1999), at 4</p> <p>[T]he parse tree represents the sentence in a hierarchical fashion, moving from a general description of the sentence (at the root of the tree) down to the specific sentence being parsed (the actual tokens) at the leaves.</p> <p>Allen I. Holub, COMPILER DESIGN IN C (1999), at 4</p> <p>A data structure containing zero or more nodes that are linked together in a hierarchical fashion. If there are any nodes, one node is the root; each node except the root is the child of one and only one other node; and each node has zero or more nodes</p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>the first tree element <b>24 a</b>. If the parser <b>14</b> determines in step <b>44</b> that the first command word is valid, the parser <b>14</b> continues searching the next command word in step <b>46</b>. If the first command word is invalid based on no match in the first element <b>24 a</b> of the command parse tree, the parser <b>14</b> returns an invalid command message to the user in step <b>56</b>.” (4:10-18)</p> <p>Figures 2, 3.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein, including:</p> <p>“Furthermore, the applicant claims that Shieber teaches that since the parse tree is rewritten several times by a rewriter, the command is not validated based on the command parse tree. However, this is untrue. The rewrite rules do validate commands on the parse tree by matching commands on the tree (identifying one of the elements) (Column 3, lines 46-49).” (File History, Rejection, March 18, 2004)</p> <p>“However, claims 1, 10, and 14 specify a <u>command parse tree</u> (claim 23 is addressed below). The command parse tree specifies <u>valid generic commands relative to a prescribed generic command format</u>. Further, the command parse tree has elements, each specifying at least one generic command component and a corresponding at least one command action value. As described in the specification at page 4, lines 16-22: ‘the parser 14 and the translators 16 provide a generic command syntax that integrates the functionality of the different tools 18 and that automatically selects the appropriate command for the best tool for executing a given generic command. As illustrated in Part A of the attached appendix, the new syntax provides a generic instruction set that provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue command based on the relative functions, as opposed to the specific syntax for a corresponding</p>	<p>as children.</p> <p>MICROSOFT COMPUTER DICTIONARY 529 (5th ed. 2002).</p> <p>A hierarchical structure like an organization chart.</p> <p>COMPUTER DESKTOP ENCYCLOPEDIA 991 (9th ed. 2001).</p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>tool 18.” (File History, Applicant Response, June 23, 2004)</p> <p>“Note, however, that even though the command word translation table 20 may not necessarily identify the <u>syntax</u> (e.g., ordered sequence of parameters relative to command words), the parser 14 includes a command parse tree having a <u>structure</u> (interconnected tree elements 24) that implements the syntax illustrated in Part B of the Appendix.” (File History, Applicant Response, June 23, 2004)</p>	
<p>“means for validating a generic command received from a user”</p> <p><u>Proposed construction:</u></p> <p><u>Function:</u> validating a generic command received from a user</p> <p><u>Structure:</u> Parser 14 in Figure 2, which includes the command word translation table 20 and the command parse tree 22, as described in 3:36-61, and equivalents</p>	<p>The specification of the ‘526 patent, including:</p> <p>“FIG. 2 is a diagram illustrating in detail the parser 14 of FIG. 1 according to an embodiment of the present invention. The parser 14 includes a command word translation table 20 and a command parse tree 22. The command word translation table 20 is configured for storing, for each prescribed command word 26, a corresponding token value 28 that is used by the parser 14 to identify a specific command for a selected one of the translators 16. In particular, the command word translation table 20 includes all the command words 26 that are valid according to the generic syntax, illustrated for example in Part B of the attached appendix.</p> <p>The parser 14 is configured for validating a received generic command by comparing each input command word to the command parse tree 22 to determine for the received generic command a tree element 24 identified as a best match. Each tree element 24 includes at least one token-command key pair 30 that specifies a token (T) 28 and a corresponding command key (CK) 32, enabling the parser 14 to identify the appropriate prescribed command based on the command key specified for the matching token. In particular, the parser 14 recursively traverses the command parse tree 22 for each command word to identify the best match for the generic command. If only a portion of the generic</p>	<p>Validate: 2. to corroborate or support on a sound basis or authority : VERIFY, SUBSTANTIATE</p> <p>Webster’s Third New International Dictionary, 2002</p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>command is identified as valid (e.g., only the first three command words are valid), the parser 14 selects the command key 32 for the matching token 28 from the last valid tree element 24.” (3:36-61)</p> <p>Figure 2.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein, including:</p> <p>“Claim 23 specifies means for validating a generic command received from a user. This means for validating reads on the disclosed structure, namely the disclosed parser 14 of Figure 2 which includes the command word translation table 20 and the command parse tree 22;” (File History, Applicant Response, June 23, 2004)</p>	
<p>“respective command formats”</p> <p><u>Proposed construction:</u> “command format specific to a management program”</p>	<p>The specification of the ‘526 patent, including:</p> <p>“There is a need for an arrangement that integrates multiple RTM programs and command and control functionality for a user, without the necessity of learning the respective command formats and syntax.” (1:41-44)</p> <p>“There is also a need for arrangement that enables a simple command language to be utilized for control of multiple RTM programs having respective command formats.” (1:45-47)</p> <p>“The parser, upon identifying a best match among the elements, issues a prescribed command for a selected one of the management programs according to the corresponding command format based on the selected command action value. Hence, a user may control multiple management programs having respective command formats, by using a set of generic commands that are independent from the command formats, eliminating the necessity that the user needs to learn the detailed command formats and syntax.” (1:54-63)</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p data-bbox="662 321 1240 579">“As illustrated in Part A of the attached appendix, the new syntax provides a generic instruction set that provides an abstraction of the tool-specific command formats and syntax, enabling a user to issue command based on the relative functions, as opposed to the specific syntax for a corresponding tool <b>18.</b>” (3:31-35)</p> <p data-bbox="662 611 1208 705">The prosecution history of the ‘526 patent, including prior art references cited therein, including:</p> <p data-bbox="662 737 1232 1119">“Hence, each of the independent claims specify that, upon identifying an element as a best match relative to the generic command, a prescribed command is issued <u>for one of the management programs</u> and according to the command format for the corresponding management program. These and other features are not disclosed in the applied reference, and as such distinguish the independent claims from the applied reference.” (File History, Applicant Response, April 14, 2003)</p> <p data-bbox="662 1150 1240 1881">“Although Pratt discloses a method and apparatus for parsing commands using a parse tree, the Official Action fails to identify the presence of any <u>management programs</u> having respective command formats, where the generic command is mapped to a <u>prescribed</u> command for a <u>selected</u> management program having a <u>corresponding command format</u>. In particular, the Official Action merely states in paragraph 2, page 2, that Pratt teaches “issuing a prescribed command based on the matched element (Column 1, lines 26-31). Claim 14 corresponds directly with Claim 1 and is rejected for the same reasons as Claim 1.” Claims 1 and 14, however, specify the distinguishing feature of issuing a prescribed command of a <u>selected</u> management program (from multiple management programs having <u>respective command formats</u>) according to the <u>corresponding command format</u>.” (File History, Applicant Response, April 14, 2003)</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>“Independent claims 1, 10, 14 and 23 are directed to issuing commands for a <u>selected one</u> of multiple management programs having respective command formats, based on a received generic command from a user. In particular, each claim specifies that a generic command is validated based on identifying, within a command parse tree, an element as a best match relative to the generic command. Each independent claim also specifies that a prescribed command for a selected one of the management programs is issued based on the identified element: in particular, claims 1 and 14 specify “issuing a prescribed command of a selected one of the management programs according to the corresponding command format, based on the identified one element”;...</p> <p>Hence, each of the independent claims specify that, upon identifying an element as a best match relative to the generic command, a prescribed command is issued <u>for one of the management programs</u> and according to the command format for the corresponding management program. These and other features are not disclosed in the applied reference, and as such distinguish the independent claims from the applied reference.” (File History, Applicant Response, Oct. 7, 2003)</p> <p>“Neither of the references, singly or in combination, address the problem of executing a <u>plurality of management programs according to respective command formats</u>, where a generic command can be converted to a prescribed command for a <u>selected management program</u>.” (File History, Applicant Response, June 18, 2004)</p> <p>Claim 1.</p>	
<p>“the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at</p>	<p>The specification of the ‘526 patent, including:</p> <p>“One aspect of the present invention provides a method in a processor-based system configured for executing a plurality</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>least one command action value”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)</p>	<p>of management programs according to respective command formats. The method includes receiving a generic command from the user, and validating the generic command based on a command parse tree that specifies valid generic commands relative to a prescribed generic command format, the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating step including identifying one of the elements as a best match relative to the generic command. “ (1:64-2:8)</p> <p>Figures 1, 2, 3.</p> <p>Claims 1, 14.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	
<p>“the validating step including identifying one of the elements as a best match relative to the generic command”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)</p>	<p>The specification of the ‘526 patent, including:</p> <p>“One aspect of the present invention provides a method in a processor-based system configured for executing a plurality of management programs according to respective command formats. The method includes receiving a generic command from the user, and validating the generic command based on a command parse tree that specifies valid generic commands relative to a prescribed generic command format, the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating step including identifying one of the elements as a best match relative to the generic command. “ (1:64-2:8)</p> <p>Figure 2.</p> <p>Claims 1, 14.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	



Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>“issuing a prescribed command of a selected one of the management programs according to the corresponding command format, based on the identified one element”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)</p>	<p>The specification of the ‘526 patent, including:</p> <p>“One aspect of the present invention provides a method in a processor-based system configured for executing a plurality of management programs according to respective command formats. The method includes receiving a generic command from the user, and validating the generic command based on a command parse tree that specifies valid generic commands relative to a prescribed generic command format, the command parse tree having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating step including identifying one of the elements as a best match relative to the generic command. “ (1:64-2:8)</p> <p>Figures 1, 2, 3.</p> <p>Claims 1, 14.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	
<p>“command word translation table, configured for storing for each prescribed command word a corresponding token”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning</p>	<p>The specification of the ‘526 patent, including:</p> <p>Figures 1, 2, 3.</p> <p>Claims 2, 15.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	
<p>“recursively traversing the command parse tree based on an order of the input command ”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)</p>	<p>The specification of the ‘526 patent, including:</p> <p>Figures 1, 2, 3.</p> <p>Claims 3, 16.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	



Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>“issuing the prescribed command based on a corresponding command key specified for the matching token”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning</p>	<p>The specification of the ‘526 patent, including:</p> <p>Figures 1, 2, 3.</p> <p>Claims 4, 17.</p> <p>The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	
<p>“command key”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning</p>	<p>The specification of the ‘526 patent, including:</p> <p>“The parser <b>14</b> is configured for validating a received generic command by comparing each input command word to the command parse tree <b>22</b> to determine for the received generic command a tree element <b>24</b> identified as a best match. Each tree element <b>24</b> includes at least one token-command key pair <b>30</b> that specifies a token (T) <b>28</b> and a corresponding command key (CK) <b>32</b>, enabling the parser <b>14</b> to identify the appropriate prescribed command based on the command key specified for the matching token. In particular, the parser <b>14</b> recursively traverses the command parse tree <b>22</b> for each command word to identify the best match for the generic command. If only a portion of the generic command is identified as valid (e.g., only the first three command words are valid), the parser <b>14</b> selects the command key <b>32</b> for the matching token <b>28</b> from the last valid tree element <b>24</b>.” (3:47-61)</p> <p>“The parser <b>14</b> then parses the next word (e.g., “tcp”) of the received generic command in step <b>46</b> by locating the corresponding token <b>28</b> (e.g., “6” for “tcp”) in the table <b>20</b>, and then traversing in step <b>48</b> the tree elements that depend from the matched tree element <b>24 a</b> (e.g., <b>24 b</b>). The parser <b>14</b> determines a match between the token <b>28</b> (“6”) corresponding to the command word “tcp” in the token-command key pair <b>30 d</b> in step <b>50</b>, enabling the parser to continue for the next command word. As described above, the parser <b>14</b> repeats the process in step <b>52</b> for the third command</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
	<p>word “connections” having the token “2” and identifying a match between the entire generic command and the token-command key <b>30</b> specified in the tree element <b>24 c</b>. The parser <b>14</b> identifies in step <b>54</b> the prescribed command for a selected one of the translators <b>16</b> based on the value of the command key <b>32</b> within the matching token-command key pair <b>30</b> (e.g., “CK=3”) of the last valid command word, which maps to a translation table that specifies a specific command for a specific translator <b>16</b>.” (4:19-36)</p> <p>“As described above, the parser <b>14</b> can identify a command key <b>32</b> even if only a portion of the command is valid. Assume for example that the parser <b>14</b> receives the invalid command “get udp connection info”. In this case, the individual command words are valid from the command word translation table <b>20</b>, however, the sequence is invalid. In particular, the command word “get” having a token value of “3” reaches the token-command key pair <b>30 b</b>, however the command word “udp” having a token value of “7” does not reach any child of the tree element <b>24 a</b>. Hence, the parser <b>14</b> uses the last valid command key (“6”) in step <b>54</b> based on the matching token for the first valid word located in the token-command key pair <b>30 b</b>. The command key is mapped to a selected one of the translators <b>16</b> in an attempt to provide a command to the corresponding resource <b>18</b>. If the selected resource <b>18</b> determines that the command is invalid, the selected resource <b>18</b> at that time may prompt the user for a correct command.” (4:37-53)</p> <p>Figures 1, 2, 3. Claims 5, 18. The prosecution history of the ‘526 patent, including prior art references cited therein.</p>	

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>“means for validating a generic command received from a user, the validating means configured for specifying valid generic commands relative to a prescribed generic command format and having elements each specifying at least one corresponding generic command component and a corresponding at least one command action value, the validating means identifying one of the elements as a best match relative to the generic command”</p> <p><u>Proposed construction:</u></p> <p>For “means for validating a generic command received from a user” see above. For rest of term, plain and ordinary meaning (except that specific terms appearing within the phrase should be construed as proposed above)</p>	<p>See “means for validating a generic command received from a user.”</p>	

## II. The '886 Patent

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>"extensible markup language (XML)"</p> <p><u>Proposed construction:</u></p> <p>"extensible": a property of a computer language that allows the user to add new features or modify existing ones</p> <p>"markup language": a computer language that allows the user to add identifiers to a document for indicating logical components or layout</p>	<p>The specification of the '886 patent, including col. 3:10-7:21.</p> <p>The prosecution history of the '886 patent, including prior art references cited therein.</p>	<p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Extensible language: a programming language which can be modified by adding new features of changing existing ones.</p> <p>Markup: the process of adding information (tags) to an electronic document that are not part of the content but describe its structure or elements.</p> <p>Markup language: a set of rules and procedures for markup.</p> <p><u>Microsoft Computer Dictionary, 5<sup>th</sup> Ed. (2002):</u></p> <p>markup language <i>n.</i> A set of codes in a text file that instructs a computer how to format the file on a printer or video display or how to index and link its contents. Examples of markup languages are Hypertext Markup Language (HTML) and Extensible Markup Language (XML), which are used in Web pages, and Standard Generalized Markup Language (SGML), which is used for typesetting and desktop publishing purposes and in electronic documents. Markup languages of this sort are designed to enable documents and other files to be platform-independent and highly portable between applications. <i>See also</i> HTML, SGML, XML.</p> <p><u>Dictionary of computer science, engineering, and technology (2001):</u></p> <p>Markup language: one of any languages for annotation of source code to simply improve the source code's appearance with the means of bold-faced keywords, slanted comments, etc. In computerized document preparation, a method of adding information to the text indicating the logical components of a document, or instructions for layout of the text on the page or other information which can be interpreted by some automatic system.</p>
<p>"command line interface (CLI)"</p>	<p>The specification of the '886 patent,</p>	<p><u>Dictionary of computer science, engineering, and technology (2001):</u></p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p>parser”</p> <p><u>Proposed construction:</u></p> <p>a component of the routing system for analyzing command line interface (CLI) commands using a grammar</p>	<p>including Abstract, cols. 1:41-64, 2:52-5:9; 6:11-24.</p> <p>The prosecution history of the ‘886 patent, including prior art references cited therein, including Amendments filed on Jan. 19, 2010, Dec. 8, 2010, and cited reference U.S. Patent No. 5,778,223.</p>	<p>Parsing: the process by which an input string is analyzed using a grammar to determine if the input string satisfies the rules of the grammar.</p> <p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Parsing: a process whereby phrases in a string of characters in a computer language are associated with the component names of the grammar that generated the string.</p> <p><u>Wiley electrical and electronics engineering dictionary (2004):</u></p> <p>Parse: 1. To examine closely and break down into components. 2. In computer, to analyze and separate into components which are more easily processed, converted, or the like.</p>
<p>“internetwork operating system (IOS) command line interface (CLI) parser subsystem”</p> <p><u>Proposed construction:</u></p> <p>a subcomponent of the internetwork operating system (IOS) for analyzing command line interface (CLI) commands using a grammar.</p>	<p>The specification of the ‘886 patent, including Abstract, cols. 1:41-64, 2:52-5:9; 6:11-24.</p> <p>The prosecution history of the ‘886 patent, including prior art references cited therein, including Amendments filed on Jan. 19, 2010, Dec. 8, 2010, and cited reference U.S. Patent No. 5,778,223.</p>	<p><u>Dictionary of computer science, engineering, and technology (2001):</u></p> <p>Parsing: the process by which an input string is analyzed using a grammar to determine if the input string satisfies the rules of the grammar.</p> <p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Parsing: a process whereby phrases in a string of characters in a computer language are associated with the component names of the grammar that generated the string.</p> <p><u>Wiley electrical and electronics engineering dictionary (2004):</u></p> <p>Parse: 1. To examine closely and break down into components. 2. In computer, to analyze and separate into components which are more easily processed, converted, or the like.</p>
<p>“XML tag”</p> <p><u>Proposed</u></p>	<p>The specification of the ‘886 patent,</p>	<p><u>Microsoft Computer Dictionary, 5<sup>th</sup> Ed. (2002):</u></p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
<p><u>construction:</u></p> <p>one or a pair of XML indicators identifying data</p> <p>“extensible markup language” or “XML” as construed above</p>	<p>including col. 4:62-6:50.</p> <p>The prosecution history of the ‘886 patent, including prior art references cited therein, including Amendment filed on Jun. 15, 2009.</p>	<p>tag <i>n</i>. <b>1.</b> In programming, one or more characters containing information about a file, record type, or other structure. <b>2.</b> In certain types of data files, a key or an address that identifies a record and its storage location in another file. <i>See also</i> tag sort. <b>3.</b> In markup languages such as SGML and HTML, a code that identifies an element in a document, such as a heading or a paragraph, for the purposes of formatting, indexing, and linking information in the document. In both SGML and HTML, a tag is generally a pair of angle brackets that contain one or more letters and numbers. Usually one pair of angle brackets is placed before an element, and another pair is placed after, to indicate where the element begins and ends. For example, in HTML, &lt;I&gt;hello world&lt;/I&gt; indicates that the phrase “hello world” should be italicized. <i>See also</i> &lt;&gt;, element, emotag, HTML, SGML. <b>4.</b> An early-generation raster graphics format used for Macintosh Ready, Set, Go programs and Letraset’s ImageStudio. <i>See also</i> raster graphics.</p> <p>markup language <i>n</i>. A set of codes in a text file that instructs a computer how to format the file on a printer or video display or how to index and link its contents. Examples of markup languages are Hypertext Markup Language (HTML) and Extensible Markup Language (XML), which are used in Web pages, and Standard Generalized Markup Language (SGML), which is used for typesetting and desktop publishing purposes and in electronic documents. Markup languages of this sort are designed to enable documents and other files to be platform-independent and highly portable between applications. <i>See also</i> HTML, SGML, XML.</p> <p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Extensible language: a programming language which can be modified by adding new features of changing existing ones.</p> <p>Markup: the process of adding information (tags) to an electronic document that are not</p>

Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
		<p>part of the content but describe its structure or elements.</p> <p>Markup language: a set of rules and procedures for markup.</p> <p><u>Dictionary of computer science, engineering, and technology (2001):</u></p> <p>Markup language: one of any languages for annotation of source code to simply improve the source code's appearance with the means of bold-faced keywords, slanted comments, etc. In computerized document preparation, a method of adding information to the text indicating the logical components of a document, or instructions for layout of the text on the page or other information which can be interpreted by some automatic system.</p>
<p>“parsing the output message to identify at least one CLI token”</p> <p><u>Proposed construction:</u></p> <p>analyzing the output message to extract at least one unit of CLI characters in a sequence</p>	<p>The specification of the ‘886 patent, including col. 6:30-7:21.</p> <p>The prosecution history of the ‘886 patent, including prior art references cited therein, including Amendments filed on Jan. 21, 2009 and Jun. 15, 2009.</p>	<p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Token: 1. A distinguishable unit in a sequence of characters. 2. A single byte that is used to represent a keyword in a programming language in order to conserve storage space. 3. A physical object, such as a badge or identity card, issued to authorized users of a computer system, building, or area.</p> <p><u>Dictionary of computer science, engineering, and technology (2001):</u></p> <p>Parsing: the process by which an input string is analyzed using a grammar to determine if the input string satisfies the rules of the grammar.</p>
<p>“receiving, with a command line interface (CLI) parser”</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning, except for “command line interface (CLI)”</p>	<p>The specification of the ‘886 patent, including cols. 3:60-4:16; 4:50-5:57; 5:66-6:10.</p> <p>The prosecution history of the ‘886 patent, including prior art references cited therein.</p>	<p><u>Dictionary of computer science, engineering, and technology (2001):</u></p> <p>Parsing: the process by which an input string is analyzed using a grammar to determine if the input string satisfies the rules of the grammar.</p> <p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Parsing: a process whereby phrases in a string of characters in a computer language are</p>



Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
parser" as construed above		<p>associated with the component names of the grammar that generated the string.</p> <p><u>Wiley electrical and electronics engineering dictionary (2004):</u></p> <p>Parse: 1. To examine closely and break down into components. 2. In computer, to analyze and separate into components which are more easily processed, converted, or the like.</p>
<p>"wherein the input command is configured in an extensible markup language (XML) format having a CLI syntax with CLI keywords sequenced according to configuration rules for CLI commands"</p> <p><u>Proposed construction:</u></p> <p>Plain and ordinary meaning, except for "extensible markup language (XML) format" as construed above</p>	<p>The specification of the '886 patent, including at col. 3:10-5:65.</p> <p>The prosecution history of the '886 patent, including prior art references cited therein.</p>	<p><u>McGraw-Hill Dictionary of Scientific and Technical Terms, 6<sup>th</sup> Ed (2003):</u></p> <p>Extensible language: a programming language which can be modified by adding new features of changing existing ones.</p> <p>Markup: the process of adding information (tags) to an electronic document that are not part of the content but describe its structure or elements.</p> <p>Markup language: a set of rules and procedures for markup.</p> <p><u>Microsoft Computer Dictionary, 5<sup>th</sup> Ed. (2002):</u></p> <p>markup language <i>n</i>. A set of codes in a text file that instructs a computer how to format the file on a printer or video display or how to index and link its contents. Examples of markup languages are Hypertext Markup Language (HTML) and Extensible Markup Language (XML), which are used in Web pages, and Standard Generalized Markup Language (SGML), which is used for typesetting and desktop publishing purposes and in electronic documents. Markup languages of this sort are designed to enable documents and other files to be platform-independent and highly portable between applications. <i>See also</i> HTML, SGML, XML.</p> <p><u>Dictionary of computer science, engineering, and technology (2001):</u></p> <p>Markup language: one of any languages for annotation of source code to simply improve the source code's appearance with the means of bold-faced keywords, slanted comments, etc. In computerized document preparation, a</p>



Claim Term & Cisco's Proposed Construction	Supporting Intrinsic Evidence	Supporting Extrinsic Evidence
		method of adding information to the text indicating the logical components of a document, or instructions for layout of the text on the page or other information which can be interpreted by some automatic system.
<p data-bbox="305 548 591 684">"XML parameter"  Proposed construction:</p> <p data-bbox="305 716 591 842">Plain and ordinary meaning, except for "XML" as construed above</p>	<p data-bbox="613 548 899 674">The specification of the '886 patent, including at col. 3:10-5:65.</p> <p data-bbox="613 716 899 863">The prosecution history of the '886 patent, including prior art references cited therein.</p>	